

PR #54403 完整报告

vllm-project/vllm

[ROCm][CI] Stabilize the sqrt-softplus top-k tie oracle

合并时间: 2026-09-01 13:34

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/54403>

执行摘要

- 一句话: 修复测试参考实现中的 oracle 模糊性, 消除 CI 间歇性失败
- 推荐动作: 该 PR 值得阅读。它展示了如何通过最小化的测试代码变更来诊断和解决由底层算子行为不确定性导致的 CI 非确定性故障。对于负责维护 MoE 路由或编写类似内核参考实现的工程师, 其中关于使用 stable argsort 来保证确定性平局打破的方法具有直接的参考价值。

功能与动机

MI355 MoE Shard 3 在 BF16 384-expert 案例下报告了一个专家不匹配。bisection 分析未找到责任 PR, 因为失败时的源码与可复现成功的版本完全一致。根本原因是测试参考实现中使用的 `torch.topk` 在分数相同时不保证选择哪个索引 (oracle 模糊), 而实际 fused kernel 故意选择了较低的专家 ID。通过使参考排序逻辑变得确定性, 消除了这一模糊性。

实现拆解

1. 修改测试辅助函数: 在 `tests/kernels/moe/test_topk_softplus_sqrt.py` 中, 修改 `_torch_topk_softplus_sqrt` 函数的 `else` 分支 (即无 `hash_indices_table` 时的路径)。将 `torch.topk` 调用替换为 `torch.argsort(scores_for_choice, dim=-1, descending=True, stable=True)[:, :topk]`。这样做的原因是 `argsort` 在 `descending=True` 且 `stable=True` 时, 对于相等分数的元素会保持其原始索引顺序 (即按专家 ID 升序), 从而实现了与 fused kernel 一致的确定性平局打破逻辑。
2. 新增单元测试: 新增函数 `test_torch_topk_softplus_sqrt_breaks_ties_by_expert_id`。该测试构造了一个简单的 gating_output, 其中包含两个分数相同的专家 (索引 1 和 2), 调用修改后的参考实现, 并断言当 `topk=2` 时, 选中的专家 ID 序列为 `[[0, 1]]`, 即分数相同的专家中, 索引更小的专家 (ID=1) 被优先选中, 验证了平局打破行为。
3. 无生产代码变更: 本次变更完全局限于测试文件, 旨在稳定测试参考实现, 使其行为与生产 fused kernel 完全对齐。生产代码、路由逻辑和内核行为均未改动。

关键文件:

- `tests/kernels/moe/test_topk_softplus_sqrt.py` (模块 Topk 路由; 类别 test; 类型 test-coverage; 符号 `_torch_topk_softplus_sqrt`, `test_torch_topk_softplus_sqrt_breaks_ties_by_expert_id`): 本次变更的唯一文件, 包含了需要修复的参考实现函数和新增的验证测试。

关键符号: `_torch_topk_softplus_sqrt`, `test_torch_topk_softplus_sqrt_breaks_ties_by_expert_id`

关键源码片段

`tests/kernels/moe/test_topk_softplus_sqrt.py`

本次变更的唯一文件，包含了需要修复的参考实现函数和新增的验证测试。

```
# 函数 _torch_topk_softplus_sqrt 的核心变更部分 (else 分支)
# 变更前:
# topk_ids = torch.topk(scores_for_choice, k=topk, dim=-1, sorted=True)[1]

# 变更后:
# 使用 argsort 并设置 stable=True 来匹配 fused kernel 的确定性 tie-break。
# 当分数相同时，stable 排序会保持原始索引（专家 ID）的顺序，即优先选择较小的 ID。
topk_ids = torch.argsort(
    scores_for_choice, dim=-1, descending=True, stable=True
)[: , :topk]

# 新增的测试函数，验证平局打破行为
def test_torch_topk_softplus_sqrt_breaks_ties_by_expert_id():
    # 构造输入，其中专家 1 和专家 2 的 gating_output 相同（均为 1.0），
    # 经过 softplus 和 sqrt 后分数也相同。
    gating_output = torch.tensor([[2.0, 1.0, 1.0, 0.0]])

    # 调用参考实现，topk=2
    _, topk_ids = _torch_topk_softplus_sqrt(
        gating_output,
        topk=2,
        renormalize=False,
        routed_scaling_factor=1.0,
    )

    # 断言：分数相同时，选择了索引较小的专家 (0, 1)，而非 (0, 2)
    assert topk_ids.tolist() == [[0, 1]]
```

评论区精华

本次 PR 的 review 讨论较为简洁。主要的交流发生在 Issue 评论中，用于触发 CI 构建。合并者 [tjtanaa](#) 对变更表示了批准 ("LGTM")。没有在 review 评论中出现技术争议。PR body 本身提供了清晰的技术解释和动机说明。

- CI 间歇性失败的根因诊断与修复 (correctness): 结论是将参考实现中的 `torch.topk` 替换为行为确定的 `torch.argsort(stable=True)`，使测试参考逻辑与生产内核对齐。

风险与影响

- 风险:

1. 低回归风险：变更仅修改了测试文件 `tests/kernels/moe/test_topk_softplus_sqrt.py` 中的参考实现和测试用例，未触及任何生产代码路径。因此，对 vLLM 运行时行为的直接影响为零。
2. 测试正确性风险：风险在于新实现的参考逻辑 (`torch.argsort`) 是否真的在所有情况下都与生产 fused kernel 的行为一致。通过新增的边界测试 (`test_torch_topk_softplus_sqrt_breaks_ties_by_expert_id`) 和 PR body 中提到的“完整参数矩阵和重复进程验证”，该风险得到了针对性覆盖。

- 影响：

1. 对用户的影响：无直接影响。生产代码和用户可见行为未发生任何变化。
2. 对系统 /CI 的影响：直接解决了 AMD CI 中一个间歇性的测试失败问题。通过消除测试 oracle 的模糊性，提高了 `tests/kernels/moe/test_topk_softplus_sqrt.py` 中参数化测试在所有平台上的稳定性和可靠性。
3. 对团队的影响：为测试工程师提供了一个清晰的模式：在编写参考实现以验证底层异构内核（如 fused MoE kernel）时，必须注意操作在边界情况下的确定性，避免依赖像 `torch.topk` 这样在平局打破行为上未做规定的算子。 - 风险标记：测试稳定性修复

关联脉络

- 暂无明显关联 PR